Content-based Detection and Prevention of Spam over IP Telephony - System Design, Prototype and First Results

Dirk Lentzen, Gary Grutzek, Heiko Knospe and Christoph Pörschmann
Cologne University of Applied Sciences, Germany
Email: d.lentzen@gesid.com,{gary.grutzek|heiko.knospe|christoph.poerschmann}@fh-koeln.de

***Abstract*—Spam over IP-telephony (SPIT) may emerge as a major threat requiring effective protection mechanisms. A number of anti-SPIT frameworks have been proposed in the last years. These are mainly based on call pattern and signaling analysis as well as caller reputation techniques resulting in black, grey or white lists of callers. Charging schemes and an extension of the call setup with challenge-response procedures have also been investigated.**

An analysis of the audio content is often claimed to be inappropriate since the call in question is already established when audio data are available. This contribution, however, shows that media data can nevertheless be effectively used for SPIT mitigation. A robust audio fingerprint of spectral feature vectors is computed for incoming audio data. Using a database of feature vectors, new calls are compared with previous ones and replays with identical or similar audio data are detected. Depending on the policy, future calls from the same source can then be blocked during call setup. A prototype based on this approach has been developed and first results show that the system can effectively detect and block Spam calls.

***Index Terms*—Spam, SPIT, Voice over IP, Audio Identification, Audio Fingerprinting.**

I. Introduction

Voice-over-IP (VoIP) telephony offers new possibilities and cost reduction but also contains the risk that telephony spam (SPIT) emerges as a considerable nuisance and problem, similar to email spam. With standard software, prerecorded audio messages (robocalls) can be automatically placed for a large number of callees.

Various protective mechanisms based on the analysis of call patterns, network signaling traffic and caller reputation have been proposed so far. It is noteworthy that content-based methods have been claimed to be inadequate for SPIT mitigation [1] since at the time the media data can be analyzed, the SPIT call has already been established. While this holds true for the first call, the result of a content-based analysis can as well be employed for a classification of callers and then provide protection against future SPIT calls.

It is desirable to detect SPIT even after a small amount of calls, which is probably possible only with the help of an audio-based analysis. Furthermore, it can be learned from the evolution of email spam that adversaries adapt to spam filters quickly. Hence SPIT filters depending solely on the fact that adversaries are unable to create calls that look like normal calls may be only of limited use in the future. Flexibility will be a key requisite in future SPIT mitigation systems, because changes in SPIT patterns to overcome filter systems are to be expected. Here the respective audio content may play an important role in order to distinguish SPIT calls from normal calls.

This paper is organized as follows: First, work related to SPIT mitigation is presented. Algorithms for computing and comparing audio fingerprints are explained in the subsequent section. Afterwards, the architecture of the SPIT protection system is presented. Finally, we are going to discuss the results with regard to effectiveness and performance of the proposed system.

II. Related Work

During the last years, a number of SPIT-mitigation methods and architectures have been proposed. The problem definition and some basic approaches are contained in RFC 5039 [1]. The suggested methods include black lists, white lists, reputation systems, Turing tests and payments at risk. Obviously authenticated caller identities, e.g. as suggested in RFC 4474 [2], play an important role. It is generally assumed that a combination of different techniques and an integrated anti-SPIT framework are necessary to deal successfully with SPIT. In the following, some relevant frameworks are described.

The signaling data and particularly the call frequency of a caller is monitored in an approach named *Progressive Multi Gray-Leveling* (PMG) [3]. A call is blocked if the sum of short and long term gray-levels of a caller lies above a certain threshold. Kolan and Dantu have proposed a Voice Spam Detector (VSD) which basically combines a presence service with statistical call data analysis [4]. In this case, callers or domains exceeding a given call rate are blocked. Additionally, the approach features a reputation system which is modelled on a number of properties reflecting rules of human interaction in social networks. Furthermore, Bayesian learning is applied to the SIP messages in order to estimate the probability of a SPIT call.

Contribution [5] proposes a combination of SPIT attack modeling and a challenge-response scheme using Captchas

from a trusted provider.

The SPIDER project [6] and the NEC SEAL [7], [8] architectures are designed to offer great flexibility with the help of different filter modules. This flexibility could be a key against rather agile SPITters, who usually adapt their methods and patterns quickly. The SEAL system, which was implemented as a prototype, contains different layers called stages to classify a call as SPIT that are based on the level of interaction between the system and its users. On the lower stages, SPIT detection based on the signaling data is implemented. The higher stages feature Turing tests through recognition of communication patterns. The system plays a "Your call is being transferred" announcement; if the caller transmits audio data above a specific level threshold then he is considered to be an automatic caller due to the fact that humans are assumed to wait for the call to connect [9]. The highest stage employs direct user feedback during and after a call to feed (for example) a reputation system.

The SPIDER Architecture consists of two layers: detection and decision layer. While the detection layer consists of various modules implementing tests to classify a call as SPIT or normal call, the decision layer controls the way in which the different modules are invoked and their results are combined to make the decision. The SPIDER project proposes the use of audio signatures [10]. This is similar to the approach in this contribution (see below), but the construction of audio fingerprints and the integration of the audio analysis differs. While our system passively monitors the incoming audio data, SPIDER and SEAL actively insert an audio message and analyze the audio data during the playback.

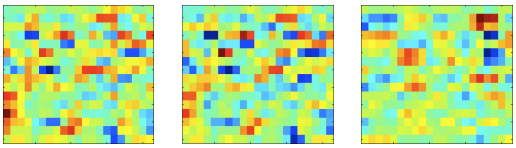
III. Audio Fingerprinting

Music identification services like $Shazam$ [11] impressively demonstrate the possibility to distinguish and recognize millions of audio samples in large databases in seconds. To accomplish this task, a compact but meaningful description of the audio material, the so-called audio fingerprint, is required. An ideal audio fingerprint allows the unique identification of an audio file in spite of alterations through telephone codecs, background noise and packet loss.

A. Feature Extraction

In the VIAT Project the fingerprint (Fig. 1) is based on Mel-Frequency Cepstrum Coefficients (MFCC) which are extracted from overlapping windowed frames of the audio sample. MFCC are computed by applying a discrete cosine transform (DCT) to the logarithmic, mel-filtered spectrum that reflects the human ears non-linear frequency perception [12]. Starting with equidistant bandpass filters up to $700Hz$, the width of the filter-banks filters logarithmically increases with frequency, thus summarizing a broader frequency range in higher filter-bank channels.

The implemented filter-bank consists of 32 bandpass filters. This results in 32 coefficients from the DCT but in our approach only the first 16 ones are considered. For channel compensation the cepstral mean is subtracted [13]. The window size is $1s$ with a $0.75s$ overlap which differs from that in standard speech processing, where $30ms$ seems to be common. Assuming that SPIT-calls from the same origin consist of temporal identical audio blocks, the huge window size distinctly reduces the amount of vectors to be compared without significantly decreasing the recognition rate.



(a) (b) (c)

Fig. 1. Fingerprints of similar (a,b) and dissimilar (c) calls

The feature extraction is done by the HTK Framework (Hidden Markov Model Toolkit [14]), a well-known frontend for speech processing.

B. Test Corpus

As a SPIT speech corpus does not exist, a suitable audio database had to be created. The audio samples originate from radio reports and audio books, converted to a 16 Bit resolution and $8kHz$ sample rate. We automatically chopped 10 hours of audio in short samples of 20 to 60 seconds. 400 files of these, which are not repeated for the whole test scenario, make up the non-SPIT Corpus. The SPIT corpus is derived from 10 other, manually chosen audio samples. To test robustness, these are altered artificially with noise, packet loss, delay, low quality G.726-codec and the full rate GSM-codec. There are 20 variants for each sample resulting in 200 files for SPIT simulation. The alterations are in a way realistic for telephone traffic but only in an exaggerated manner, while intentional degradations, like voice conversion are not considered.
The variants in detail are:

- white noise with a SNR of $15dB$, $20dB$ and $25dB$,
- pink noise with a SNR of $15dB$, $20dB$ and $25dB$,
- 5% and 10% packet loss,
- full rate GSM-codec,
- G.726-codec, $16kB/s$,
- 50, 100, 150, 250 and $500ms$ delay,
- white or pink noise with $100ms$ delay,
- white or pink noise with $250ms$ delay

The test corpus is not yet based on actual SPIT samples. The choice of the corpus should not have a significant effect on the test results, since the spit detection is solely based on the identification of repeated speech signals while the content is completely ignored. However, future tests shall be conducted with real SPIT samples.

C. Feature Comparison

The similarity of two audio samples results out of the sum of the minimum distances of the respective fingerprints vectors. The minimum distance d_{min} is calculated by assigning each

vector v of fingerprint $V = \{v_1, \ldots, v_n\}$ to the vector with the closest match of fingerprint $W = \{w_1, \ldots, w_m\}$ (Eq.1). The sum of these minimum distances is used as the distance $d(V, W)$ between two fingerprints (Eq.2).

$$d_{min}(v, W) = \min_{i=1..m} |v - w_i| \quad (1)$$

$$d(V, W) = \sum_{j=1}^{n} d_{min}(v_j, W) \quad (2)$$

For every incoming call any vector of its fingerprint has to be compared to any vector of any other call's fingerprint in the database. Therefore the comparison has to be as efficient as possible. The L^1-distance has been appointed as the most simple, yet sufficient one.

The distance is not commutative, i.e. $d(V, W) \neq d(W, V)$. Nevertheless further investigations have shown that the recognition rate does not suffer from solely comparing in one direction.

IV. System Design

The system design is based on the general concept described in a previous paper [15]. The system is intended to be a research prototype. Since the Session Initiation Protocol has widely been adopted in VoIP infrastructures, the prototype is implemented using SIP components. However, the system is not principally restricted to SIP-based IP telephony.
The design has three major goals:

- Testing Environment: The system should allow for testing and comparing the results of the different fingerprinting and matching algorithms.
- Flexibility: The system can be used for various fingerprinting methods and test cases without the need to substantially changing it.
- Validity: It should be possible to give approximate results concerning effectiveness and performance of a production system implementing this method.

Open source software is used in order to be able to make any adjustment that might be necessary. The architecture is depicted in Fig. 2. Two voice communications servers (Asterisk A and B), which generate and terminate test calls, form the basis of our system. Asterisk's call file interface is used, which generates a single call for every file in the outgoing folder. We implemented a script-based traffic generator which plays complex call scenarios specific to our requirements. Additional calls can be placed by using the SIP phones registered at Asterisk A and B. The SIP express router (SER) acts as a SIP proxy in our system. If the caller identification (URI) is black-listed and not white-listed in the PostgreSQL database, the call setup is blocked with a SIP 603 message.
Additionally, a SIP-analyzing appliance is connected to a monitor port and allows us to monitor any SIP segments and all calls on the network passively.

In Fig. 2, the components required to implement content-based SPIT detection and prevention are depicted. First, audio data of the first 60 seconds of incoming audio data at Asterisk

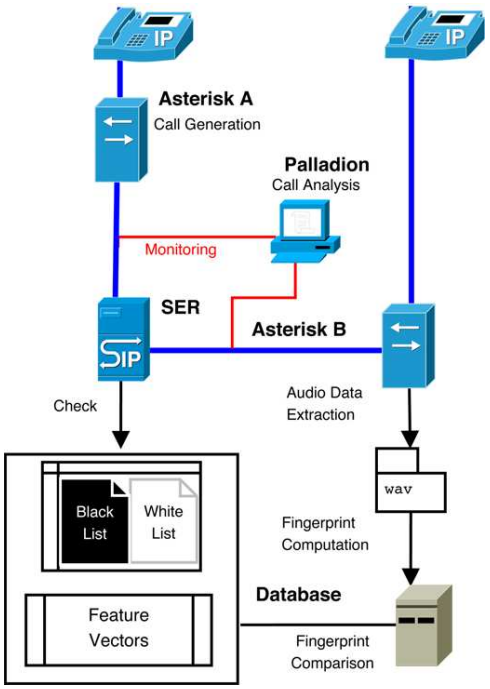


Fig. 2. System Architecture

B are written as a wav-file to a RAM disk. Then the audio fingerprint is calculated for the first 6 seconds of that particular call. It is expected that SPITters try to pass on their message early in the call and we therefore assume that the call can be uniquely characterised during the first seconds. This restriction of the analysis to the first 6 seconds should not affect the SPIT detection. The extracted feature vectors of the incoming audio data of each call are stored in a database along with some metadata, e.g. the caller URI provided in the SIP From-header. After this, the recording of the call is deleted from the RAM disk. For privacy reasons, the audio data is not permanently stored.

The main database tables are *caller*, *call* and *feature*. Information stored in the *caller* table identifies a caller by display name and the URI in the SIP From header. All calls are stored in the *call* table. Each feature vector is stored as a single row in the *feature* table. The feature vectors of one specific call all reference the same entry in the call table. This system is designed to be flexible with respect to the fingerprinting algorithm. The *feature* table possesses an attribute which specifies the type of algorithm. Thus, fingerprints from different algorithms can be processed within the same database. Via stored procedures the processing of the fingerprints is performed. These procedures consist of a wrapper layer that calls the corresponding function for the particular type of fingerprint.
When the complete fingerprint of a new call is stored in the database, the comparison with existing fingerprints is started as described in section III-C. After comparing the new fingerprint with all other fingerprints up to a configurable age, the call is marked as processed. When a newly inserted fingerprint is

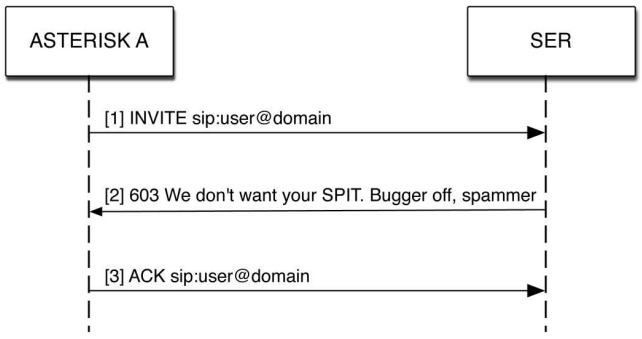


Fig. 3. Message flow for SPIT calls

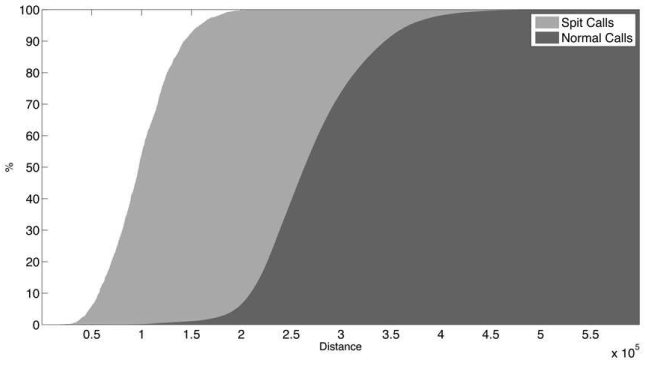


Fig. 4. Distribution of fingerprint distances

sufficiently close to a processed one, the call is probably a replay with identical or similar audio data. Depending on the policy, the caller of the last call or the caller of all matching calls are put on the black list immediately or after a number of positive recognitions. They can not place further calls in the system using that identification, provided they are not white-listed.

Subsequent SPIT calls can be reduced with minimal usage of computing power and network bandwidth. During a normal call setup at least 6 SIP messages plus provisional responses have to be relayed via the network. Fig. 3 shows the typical message flow for a SPIT call. Only three messages are exchanged between border proxy and caller. The 6xx error message indicates that the request has not been completed successfully. Further user information on the type of error could be given here. Alternatively, a Turing test or any other method to distinguish between SPIT and normal calls could be integrated.

Since the blocking of known SPIT callers can be performed at a border SIP proxy (SER in our setup), the call server (Asterisk in our setup) and the end-user can be effectively protected from a large number of Spam calls.

V. Analysis

To test the system we made VoIP calls using the set of audio files mentioned above. This leads to 600 fingerprints in the database to compare. The complete analysis of these calls requires $\frac{600}{2} \cdot 599 = 179,700$ feature comparisons (III-C). The cross comparison took 30 minutes on a virtual database server running on two cores with 8 GB RAM of an AMD 6-core opteron 2,4GHz machine. This corresponds to $10ms$ per comparison. The call generation and recording performance of our setup was at 450 parallel calls. Asterisk A and B were virtual servers with one 2,4 GHz Opteron CPU and 1GB of available RAM each.

Fig. 4 shows the resulting distribution of the sum of minimal distances of both audio file groups. Apparently the distribution of the distances of files with different content and the distribution of distances between variations of the same files show little overlap. Choosing a distance threshold of 185,000 would for example result in a SPIT recognition rate of 99% with a false positive rate of 3,2%. On the other hand, a false positive rate of 1% would still result in a recognition rate of 92%. Depending on the policy, a number of positive recognitions would be necessary before a caller is blacklisted.

VI. Future work

The proposed approach assumes that a caller can be identified with reasonable probability. This should at least be a valid assumption for VoIP carriers who charge customers for their services. Authenticated identities like those proposed in RFC 4474 [2] could be used to increase the confidence level of caller identification. The proposed system still requires a signaling/ media proxy and an interface to extract the media data as it is provided by Asterisk or commercial session border controllers (SBC). To overcome this dependency, a network based passive call extraction mechanism is preferable. A prototype already exists, but is not yet integrated into the system.

The efficient comparison of a given fingerprint to the other fingerprints in the database is still subject to further research. Studies with dynamic time warping algorithms [16] show better results but at much higher computational cost. Pre-selection of calls, indexing methods [17] and short signatures are feasible approaches which are still to be investigated. A first implementation using an inverted index shows promising results; the identification of SPIT among a set of 600 test calls was more than a hundred times faster compared to the full feature vector comparision described above. The inverted lists also scale up better with increasing telephone traffic.The feasibility and performance of this approach still has to be evaluated with a larger corpus and more simultaneous calls.

In order to enhance the system's capabilities it is planned to investigate further SPIT detection methods including keyword spotting, speech recognition and speaker identification. The system would then also need to inspect the audio content and nevertheless protect the caller's and the callee's privacy. Thus the data stored by the system must not allow a reconstruction of the audio contents. In order to achieve this objective, methods of *privacy preserving audio content matching* need to be investigated and implemented.

VII. Conclusion

While SPIT is currently not an urgent problem, it is rather certain that the abuse of telephony for unsolicited bulk messages will become relevant. Due to the synchronous communication, the nuisance of SPIT calls can be even greater than email spam. We proposed a new audio content-based method that can help to mitigate SPIT in the future. While a number of signaling-based anti-SPIT methods and frameworks have already been proposed, the research on content-based schemes is still sparse. We have designed a system that combines the advantages of audio content-based SPIT detection and signaling-based SPIT prevention. This has been achieved by calculating and comparing spectral audio fingerprints and detecting SPIT calls with identical or similar voice data. The result of the fingerprint comparison is used to generate additional black list entries. The proposed system can be integrated into existing VoIP environments and SPIT filter systems. The call data can be extracted and processed to generate the spectral feature vectors at any applicable network component where the signaling and the media data is available. The processing can be done asynchronously without affecting the existing infrastructure or manipulating the usual call flow.

By implementing a working prototype, we have shown that content-based methods are well feasible for SPIT prevention While the performance of the system still has to be improved, the results show that further research in this area is sensible.

Acknowledgment

This work has been conducted within the research project VIAT, which is supported by the German Federal Ministry of Education and Research (BMBF), reference 1736X09.

References

[1] J. Rosenberg and C. Jennings, "The Session Initiation Protocol (SIP) and Spam," RFC 5039 (Informational), Internet Engineering Task Force, Jan. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5039.txt

[2] J. Peterson and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," RFC 4474 (Proposed Standard), Internet Engineering Task Force, Aug. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4474.txt

[3] D. Shin, J. Ahn, and C. Shim, "Progressive multi gray-leveling: a voice spam protection algorithm," *IEEE Network*, vol. 20, no. 5, pp. 18–24, 2006.

[4] P. Kolan and R. Dantu, "Socio-technical defense against voice spamming," *Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 2, no. 1, 2007.

[5] D. Gritzalis and Y. Mallios, "A sip-oriented spit management framework," *Computers & Security*, vol. 27, no. 5-6, pp. 136–153, 2008.

[6] Y. Rebahi, S. Dritsas, T. Golubenco, B. Pannier, and J. F. Juell, "A conceptual architecture for SPIT mitigation," in *SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol*, S. A. Ahson and M. Ilyas, Eds. Boca Raton, FL, USA: CRC Press, Inc., 2008, ch. 23, pp. 563–580.

[7] J. Seedorf, N. d'Heureuse, S. Niccolini, and T. Ewald, "VoIP SEAL: a research prototype for protecting Voice-over-IP networks and users," in *Konferenzband der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V.(GI)*, A. Alkassar and J. Siekmann, Eds., Apr. 2008.

[8] N. d'Heureuse, J. Seedorf, and S. Niccolini, "A policy framework for personalized and role-based spit prevention," in *IPTComm '09: Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications*. New York, NY, USA: ACM, 2009, pp. 1–11.

[9] J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, and T. Ewald, "Detecting SPIT calls by checking human communication patterns," in *Proceedings of the IEEE International Conference on Communications, 2007. ICC '07*, Jun. 2007, pp. 1979–1984. [Online]. Available: ftp://lenst.det.unifi.it/pub/LenLar/proceedings/2007/ICC2007/DATA/S05S04P05.PDF

[10] Y. Rebahi, S. Ehlert, and A. Bergmann, "A spit detection mechanism based on audio analysis," in *Proceedings of 4th International Mobile Multimedia Communications Conference MobiMedia 2008: July 7-8, 2008, Oulu, Finland*. ICST; ACM, 2008.

[11] A. L. Wang, "An industrial-strength audio search algorithm," Oct 2003, pp. 7–13. [Online]. Available: http://www.ee.columbia.edu/%5C~dpwe/papers/Wang03-shazam.pdf

[12] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, Feb 1937.

[13] B. Pfister and T. Kaufmann, "Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung," p. 487, Oct 2008.

[14] Cambridge University Engineering Department. Hidden Markov Model Tool Kit. [Online]. Available: http://htk.eng.cam.ac.uk/

[15] C. Pörschmann and H. Knospe, "Spectral analysis of audio signals for the identification of spam over IP telephony," in *Proceedings of the NAG/DAGA 2009. NAG/DAGA International Conference on Acoustics, 23.-26. March 2009, Rotterdam, Niederlande*, 2009.

[16] L. R. Rabiner, "Considerations in dynamic time warping algorithms for discrete word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 6, pp. 575 – 582, 1978.

[17] F. Kurth and M. Mueller, "Efficient index-based audio matching," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 382 – 395, 2008.